

电力系统机组组合问题的动态双种群粒子群算法

李丹¹, 高立群¹, 王珂², 黄越¹

(1. 东北大学 信息科学与工程学院, 沈阳 110004; 2. 辽宁省电力有限公司, 沈阳 110000)

(lidanneu@163.com)

摘要:针对标准粒子群优化算法易陷入局部最优点的缺点,提出了动态双种群粒子群优化算法(DDPSO)。该算法中两个子种群规模随进化过程不断变化,进化中分别采用不同的学习策略且相互交换信息。将该算法应用于机组组合问题中,采用实数矩阵编码方法对发电计划进行编码,将两层优化问题转化为单层优化问题,直接运用DDPSO算法求解。仿真结果表明,用该方法解决机组组合问题具有良好的精度和鲁棒性。

关键词:粒子群优化;动态双种群;学习策略;机组组合

中图分类号: TP18 **文献标志码:** A

Dynamic double-population particle swarm optimization algorithm for power system unit commitment

LI Dan¹, GAO Li-qun¹, WANG Ke², HUANG Yue¹

(1. School of Information Science & Engineering, Northeastern University, Shenyang Liaoning 110004, China;

2. Liaoning Electric Power Company Limited, Shenyang Liaoning 110000, China)

Abstract: Dynamic Double-population Particle Swarm Optimization (DDPSO) algorithm was presented to solve the problem that the standard PSO algorithm easily fell into a locally optimized point, where the population was divided into two sub-populations varying with their own evolutionary learning strategies and exchanging between them. The algorithm had been applied to power system Unit Commitment (UC). The DDPSO particle consisted of a two-dimensional real number matrix representing the generation schedule. According to the proposed coding manner, the DDPSO algorithm could directly solve UC. Simulation results show that the proposed method performs better in term of precision and convergence property.

Key words: particle swarm optimization; dynamic double-population; learning strategy; unit commitment

0 引言

电力系统机组组合问题是典型的大规模非线性混合整数规划问题,其优化目标是:在满足系统负荷需求、旋转备用和发电机组运行技术要求等约束条件的情况下,合理地安排发电机组启停和各个时段中的机组出力,使整个系统的发电费用最小。近年来,研究人员对机组组合问题做了大量的研究工作并提出了许多解决方法,如:动态规划法、禁忌搜索法、遗传算法、人工免疫算法、粒子群算法^[1]等。上述方法虽然具体处理方式不同,但是基本上都是将机组组合问题分解成机组启停安排和经济负荷分配两层优化问题进行求解,约束条件基本上都是通过惩罚函数法来解决。此类方法由于惩罚因子的调整比较困难,所以在一定程度上影响了优化的结果。粒子群优化(PSO)算法本质是一种多代理算法,研究由简单个体组成的群落与环境以及个体之间的互动行为,具有较高的搜索效率。但是PSO算法在运行当中易陷入局部最优,出现早熟收敛现象,这是PSO算法在应用中的一个主要问题。

针对上述问题,本文提出了解决电力系统机组组合问题的动态双种群粒子群优化(DDPSO)算法,以提高PSO算法的全局搜索能力。采用二维实数矩阵对所要制定的发电计划进行编码,从而无须将机组组合问题分解成两层优化问题进行

求解。针对以往文献中惩罚因子难以调整的问题,本文提出了一种个体修正策略,通过在每次进化后对违反约束条件的个体进行修正,确保粒子群中的个体在进化更新后满足全部约束条件,保证优化结果的有效性与可行性。最后,将该方法应用于实际算例中,与其他方法的比较结果表明DDPSO算法用于解决机组组合问题具有更好的收敛速度和更高的精度。

1 机组组合问题的数学模型

机组组合问题的优化目标是在给定周期内通过发电机组的启停和出力安排使总发电成本最小。其中总发电成本包括发电机组启停费用和发电燃料费用。机组组合问题的目标函数为^[2]:

$$\min T_c = \sum_{i=1}^T \sum_{t=1}^N \{u_i(t)F_{Gi}(t) + S_{Gi}(t)\} \quad (1)$$

其中: T_c 为发电总成本; N 为发电机组总台数; T 为时段总数; $u_i(t)$ 为发电机组 i 在 t 时段的机组状态,1表示启动,0表示停机; $F_{Gi}(t)$ 为发电机组 i 在 t 时段中的发电费用(单位\$); $S_{Gi}(t)$ 为发电机组 i 在 t 时段中的机组启停费用(单位\$); i 为发电机组序号; t 为时段序号。

式(1)中发电机组的燃料费用采用二次函数形式表示为:

收稿日期:2007-07-17;修订日期:2007-09-21。

基金项目:国家自然科学基金资助项目(60274009);教育部博士点基金资助项目(20020145007)。

作者简介:李丹(1977-),男,辽宁丹东人,博士研究生,主要研究方向:智能优化;高立群(1949-),男,辽宁沈阳人,教授,博士生导师,主要研究方向:智能优化、模式识别;王珂(1957-),男,辽宁沈阳人,高级工程师,博士,主要研究方向:智能优化;黄越(1978-),女,黑龙江哈尔滨人,硕士研究生,主要研究方向:智能优化。

$$F_{Gi}(t) = a_i + b_i P_i(t) + c_i P_i^2(t) \quad (2)$$

其中: a_i, b_i, c_i 分别为发电机组 i 成本函数中常数项和一次项、二次项系数; $P_i(t)$ 为发电机组 i 在时段 t 中的发电量(单位 MW)。

机组组合问题的约束条件为:

$$\sum_{i=1}^N u_i(t) P_i(t) = D(t) \quad (3)$$

$$\sum_{i=1}^N u_i(t) P_i^{\max} \geq D(t) + R(t) \quad (4)$$

$$P_i^{\min} \leq P_i(t) \leq P_i^{\max} \quad (5)$$

$$P_i(t-1) - D_{Ri} \leq P_i(t) \leq P_i(t-1) + D_{Ri} \quad (6)$$

$$T_i^{\text{on}} \leq X_i^{\text{on}} \quad (7)$$

$$T_i^{\text{off}} \leq X_i^{\text{off}} \quad (8)$$

其中: $D(t)$ 为时段 t 中系统的负荷需求(单位 MW); P_i^{\max} 为发电机组 i 的发电能力上限(单位 MW); $R(t)$ 为时段 t 中系统旋转备用容量(单位 MW); P_i^{\min} 为发电机组 i 的发电能力下限(单位 MW); D_{Ri} 为发电机组 i 的最大降负荷速率(单位 MW/h); U_{Ri} 为发电机组 i 的最大升负荷速率(单位 MW/h); T_i^{on} 为发电机组 i 的最小连续运行时间(单位 h); X_i^{on} 为发电机组 i 的连续运行时间(单位 h); T_i^{off} 为发电机组 i 的最小连续停运时间(单位 h); X_i^{off} 为发电机组 i 的连续停机时间(单位 h)。

约束条件中:式(3)为功率平衡约束;式(4)为旋转备用约束;式(5)为机组发电能力约束;式(6)为机组爬坡速率约束;式(7)为机组最小连续运行时间约束;式(8)为机组最小连续停运时间约束。

2 动态双种群粒子群优化算法

2.1 标准 PSO 算法

由文献[3]可知,每个粒子根据如下的公式来更新自己的速度和在解空间的位置:

$$v_{id}^{(t+1)} = w v_{id}^{(t)} + c_1 r_1 (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 r_2 (p_{gd}^{(t)} - x_{id}^{(t)}) \quad (9)$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \quad (10)$$

其中: $d = 1, 2, \dots, n$; $i = 1, 2, \dots, m$, m 为种群规模; t 为当前进化代数; r_1 和 r_2 为均匀分布于 $[0, 1]$ 的随机数; c_1 和 c_2 为加速常数。

2.2 PSO 算法的学习策略

由标准 PSO 算法可知,粒子通过跟踪自己迄今为止所找到的最优解和种群迄今为止所找到最优解这两个极值来更新自己的速度,从而更新自己的位置。这种行为也可以理解为:粒子在借鉴自身和整个群体所取得的成功经验,通过对以往的成功经验的学习获得有用的信息,指导自己下一步的行动策略。同时,粒子自身和整个种群的失败经历也可以为粒子提供有意义的借鉴,使种群在后代的进化中朝着有利的方向进行。基于这一点,提出了新的 PSO 算法学习策略,这就是从以往的失败中获得有价值的信息,使粒子远离粒子本身和整个群体所找到的最差的位置,从而更新粒子的速度和位置。粒子在搜索过程中的失败可以表现为搜索到具有较差适应值的位置,记第 i 个粒子迄今为止搜索到的最差位置为 $s_i = (s_{i1}, s_{i2}, \dots, s_{in})$,整个粒子群迄今为止搜索到的最差位置为 $s_g = (s_{g1}, s_{g2}, \dots, s_{gn})$,则第 i 个粒子的速度和位置更新公式如下:

$$v_{id}^{(t+1)} = w v_{id}^{(t)} + c_1 r_1 (x_{id}^{(t)} - s_{id}^{(t)}) + c_2 r_2 (x_{id}^{(t)} - s_{gd}^{(t)}) \quad (11)$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \quad (12)$$

如果只利用式(11)和式(12)对粒子的速度和位置进行更新,也就是说只是从失败中获取经验,这与实际的经验不符。一般地,还是更多地从成功的经历中获取信息,而从失败的经历中获得相对少的信息,基于这一点在本文的算法中同时从成功和失败的经历中获取信息来更新粒子。

2.3 惯性权重的自适应调节

为了平衡种群的探测与开发能力,本文提出了基于种群多样性测度函数的惯性权重自适应调整策略。

本文采用种群多样性指数来描述种群的多样性。种群多样性指数 $D(t)$ 定义如下^[4]:

$$D(t) = \frac{f_{\min}(t)}{\sum_{i=1}^m f_i(t)} \quad (13)$$

其中: $f_i(t)$ 为在第 t 代时第 i 个粒子的适应值, $f_{\min}(t)$ 为第 t 代时所有粒子中适应值的最小值。显然, $0 < D(t) \leq 1$, $D(t)$ 越接近于 1, 表示种群越集中。

研究表明,较大的 w 值有利于提高算法的收敛速度,而 w 较小时则有利于提高算法的收敛精度。本文提出了一种按式(14)对 w 进行自适应调整的策略,即在寻优的初期,为了增加算法的全局搜索能力,惯性权重随种群多样性的增加而增大;在寻优的后期,为了增加算法的局部搜索能力,惯性权重随种群多样性的减少而减小。充分利用种群多样性的变化信息,随着种群多样性的变化非线性地调整 w 值。

$$w(t) = e^{-2D(t)} \quad (14)$$

2.4 引入变异算子和交换算子

为了保持粒子群的多样性,本文在算法中引入变异算子和交叉算子,本文采用的变异策略为,当种群多样性指数 $D(t)$ 大于某个给定的值 D_{\max} 后,按照给定的变异率 p_m 进行变异,变异时保留粒子群的历史最优位置,将要变异的粒子的速度向量中的少部分分量重新随机初始化,以此来增强全局搜索能力,同时又不降低收敛速度和搜索精度。对变异后的种群再按照给定的交叉率 p_c 进行粒子位置交换,生成新的种群,计算新种群的种群多样性指数。如果此时 $D(t)$ 小于给定的值 D_{\min} , 则进入新一轮寻优搜索,否则对种群继续进行变异和交叉操作直至 $D(t)$ 小于给定的值 D_{\min} 为止。

2.5 DDPSO 算法

本文将一个粒子群分成两个子种群,每个子种群选用不同的学习策略,即第一个子种群中的粒子选用从成功经历中获得学习信息的策略;第二个子种群中的粒子选用从失败的经历中获得学习信息的策略。可以设置一个比例系数 ρ 来控制两个子种群中粒子的个数。

$$\rho = \frac{m_2}{m_1}; m_1 + m_2 = m \quad (15)$$

其中: m 为粒子群中的粒子总数; m_1 为第一个子种群中的粒子个数; m_2 为第二个子种群中的粒子个数。

在进化的不同阶段基于两种学习策略所获得的有价值的信息量是不同的,因而两个子种群的种群规模也应随之变化,即每隔一定的代数后将整个种群按照比例系数 ρ 重新随机划分成两个子种群。本文将 ρ 设为一个非线性变化的量,其值由下式确定:

$$\rho = \rho_{\max} - (\rho_{\max} - \rho_{\min}) \frac{1 - \frac{\text{iter}_c}{0.8 \times \text{iter}_{\max}}}{1 + 0.95 \times \frac{\text{iter}_c}{0.8 \times \text{iter}_{\max}}} \quad (16)$$

其中: ρ_{\max} 和 ρ_{\min} 分别是 ρ 的最大值和最小值, $iter_c$ 和 $iter_{\max}$ 分别是种群重新划分时的进化代数和最大进化代数。

DDPSO 算法的实现步骤如下:

1) 设定 DDPSO 种群规模 m , 加速常数 c_1 和 c_2 , 比例系数 ρ 的最大值和最小值 ρ_{\max} , ρ_{\min} , 种群重新划分代数 $iter_c$, 最大进化代数 $iter_{\max}$ 。将当前进化代数设置为 $iter = 1$;

2) 在解空间中初始化粒子的速度和位置;

3) 若当前进化代数为种群重新划分代数, 按式(16)更新比例系数 ρ , 则将种群按照比例系数 ρ 划分为两个子种群; 否则转至第4步;

4) 第一个子种群按式(9)、(10)和(14)更新粒子速度和位置, 第二个子种群按式(11)、(12)和(14)更新子种群中的粒子, 并进行交叉和变异操作从而产生新种群 X_i ;

5) 评价种群 X_i 。将第 i 个粒子当前点适应值与该粒子迄今找到的最优位置 p_i (最差位置 s_i) 的适应值进行比较, 若更优(差), 则更新 p_i (s_i), 否则保持 p_i (s_i) 不变; 再与种群迄今找到的最优位置 p_g (最差位置 s_g) 的适应值进行比较, 若更优(差), 则更新 p_g (s_g); 否则保持 p_g (s_g) 不变;

6) 检查是否满足寻优结束条件, 若满足则结束寻优, 求出最优解; 否则, 置 $iter = iter + 1$, 转至第3步; 结束条件为寻优达到最大进化代数 $iter_{\max}$ 。

3 基于 DDPSO 算法的机组组合问题求解

3.1 机组组合问题的编码方法

在 DDPSO 算法中, 将各时段内所有机组的发电量串接起来作为 DDPSO 算法中的一个个体, 用一个 $N \times T$ 的矩阵代表种群中的一个个体。其位置矩阵的具体形式为:

$$P_k = (Q_1, Q_2, \dots, Q_t, \dots, Q_T) = (R_1, R_2, \dots, R_i, \dots, R_N)^T = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,t} & \dots & P_{1,T} \\ P_{2,1} & P_{2,2} & \dots & P_{2,t} & \dots & P_{2,T} \\ \vdots & \vdots & & \vdots & & \vdots \\ P_{i,1} & P_{i,2} & \dots & P_{i,t} & \dots & P_{i,T} \\ \vdots & \vdots & & \vdots & & \vdots \\ P_{N,1} & P_{N,2} & \dots & P_{N,t} & \dots & P_{N,T} \end{bmatrix} \quad (17)$$

其中: P_k 为粒子群中的第 k 个个体; $P_{i,t}$ 为编码矩阵中的第 i 行、第 t 列元素, 含义为发电机组 i 在时段 t 中的发电量大小; Q_t 为编码矩阵中的第 t 个列向量, 表示调度周期某一时段所有机组的组合状态, 可以求解经济负荷分配并计算生产成本; R_i 为编码矩阵中的第 i 个行向量, 表示某个机组在整个调度周期 T 个时段的开停机状态, 行向量必须满足机组最小开停机时间的约束, 可以计算机组的启停费用。发电机组的运行状态取决于矩阵中元素的具体取值, 即根据机组在某时段的出力大小来确定启停状态, 具体表达式为:

$$u_i(t) = \begin{cases} 0, & P_{i,t} = 0 \\ 1, & \text{其他} \end{cases} \quad (18)$$

由粒子群算法可知, 还需设定一个个体的速度矩阵, 具体形式如下:

$$V_k = \begin{bmatrix} V_{1,1} & V_{1,2} & \dots & V_{1,t} & \dots & V_{1,T} \\ V_{2,1} & V_{2,2} & \dots & V_{2,t} & \dots & V_{2,T} \\ \vdots & \vdots & & \vdots & & \vdots \\ V_{i,1} & V_{i,2} & \dots & V_{i,t} & \dots & V_{i,T} \\ \vdots & \vdots & & \vdots & & \vdots \\ V_{N,1} & V_{N,2} & \dots & V_{N,t} & \dots & V_{N,T} \end{bmatrix} \quad (19)$$

其中: $V_{i,t}$ 为粒子群位置矩阵中分量 $P_{i,t}$ 所对应的速度分量。

3.2 机组组合问题的适应度函数

本文直接采用机组组合问题的目标函数作为 DDPSO 算法的适应度函数。适应度最低的粒子运行费用最低, 即为所求的最优解。

3.3 DDPSO 种群初始化

DDPSO 种群初始化分为位置初始化和速度初始化, 下面将分别加以介绍。

1) 位置初始化

和文献[2]中的遗传算法种群初始化相类似, DDPSO 种群位置初始化过程如下:

(1) 为 P_k 矩阵中每一列生成随机数数组 $rand = (rand_1, \dots, rand_j, \dots, rand_N)$, $j = 1, 2, \dots, N$ 。

(2) 计算百分比系数数组 $per = (per_1, \dots, per_j, \dots, per_N)$, $per_j = \frac{rand_j}{\sum_{k=1}^N rand_k}$, $j = 1, 2, \dots, N$ 。

(3) 初始化各台发电机的发电量, $Q_i = (P_{1,t}, P_{2,t}, \dots, P_{j,t}, \dots, P_{N,t})^T$, $P_{j,t} = per_j D(t)$, $j = 1, 2, \dots, N$ 。

2) 速度初始化

由 DDPSO 算法可知, 各台机组的速度分量恰好为相邻的两个时段内各台机组发电量之差, 由式(6)所示的机组爬坡速率约束可知, 相邻的两个时段内各台机组发电量之差的最大值为 D_{Ri} , 所以速度分量应小于 D_{Ri} , 而各台机器的启停要满足式(7)和(8)的约束, 为了使种群中的各粒子根据式(9)~(12)更新时始终满足启停约束, 速度分量设为 $-\frac{P_i^{\min}}{T_i} \leq$

$V_{i,t} \leq \frac{P_i^{\min}}{T_i}$, $T_i = \max(T_i^{\text{on}}, T_i^{\text{off}})$, 以同时满足上述约束。则速度

初始化为 $V_{i,t} = -\frac{P_i^{\min}}{T_i} + 2rand \frac{P_i^{\min}}{T_i}$, $i = 1, 2, \dots, N, t = 1, 2, \dots, T, rand \in [0, 1]$ 。

3.4 个体调整策略

DDPSO 种群中的个体经过初始化或者经过更新后, 有可能不完全满足约束条件, 此时需要对个体进行调整, 使其满足全部约束条件, 具体的调整步骤如下:

1) 调整位置矩阵中的分量, 使其满足发电机组的发电能力约束。方法如下:

$$P_{i,t} = \begin{cases} P_i^{\max}, & P_{i,t} \geq P_i^{\max} \\ P_{i,t}, & P_i^{\min} \leq P_{i,t} < P_i^{\max} \\ P_i^{\min}, & (1 - \frac{1}{T_i})P_i^{\min} \leq P_{i,t} < P_i^{\min} \\ 0, & \text{其他} \end{cases}$$

2) 调整发电机组启停状态满足系统的旋转备用约束。若所有状态为1(开机)的机组的最大技术出力之和小于当前时刻的总负荷与旋转备用之和, 则需要按优先顺序依次将状态为0(停机)且满足最小停运时间约束的机组开机, 直至满足系统的旋转备用约束。机组的优先顺序按文献[5]中的方法确定, 即平均成本微增率较低且启动成本较高的机组的优先级别较高, 反之较低。

3) 负荷分配的调整。某一时段中所有发电机组的发电总和可能不等于系统该时段的总负荷, 因此需要进行负荷分配的调整, 调整方法如下: 若所有状态为1的机组的最小技术

出力之和大于当前时刻的总负荷,则需要按照优先顺序将状态为 1 且满足最小运行时间约束的机组停机;若当前时刻的发电量之和小于系统负荷,通过增加运行发电机组出力来满足功率平衡约束;若当前时刻的发电量之和大于系统负荷,则降低运行发电机组的出力。在调整中必须要满足机组的爬坡约束和各机组的发电能力且不能改变机组的运行状态。

经过上面的个体调整后,在理论上可能会出现如下情况:在某个调度时刻,所有能够开机的机组的最大技术出力之和仍然小于当前时刻的总负荷与旋转备用之和。这是因为在之前的时段将某些机组停机导致当前时刻虽然负荷增长较大,但由于启停时间约束的限制而无法将其开机造成的。此时的调整方法为对该时段之前的某些时段中的停机行为进行重新评估,在满足功率平衡约束的前提下以降低某些机组的技术出力来代替停机行为,使后续调度时刻有足够旋转备用。

4 算例分析

为验证本文提出方法的有效性,对 10 台机组、24 时段的算例进行仿真计算。单线图、线路参数、各发电机参数和各时段的负荷参数见文献[6]。DDPSO 算法的参数为: $m = 50$; $\rho_{\max} = 1$; $\rho_{\min} = 0$; $c_1 = c_2 = 2$; $iter_c = 10$; $iter_{\max} = 600$; $D_{\max} = 0.8$; $D_{\min} = 0.35$; $p_c = 0.6$; $p_m = 0.01$ 。

表 1 为 DDPSO 算法的优化结果,运行费用为 \$559 312。表中的非零数字为机组的实际发电量,“0”表示机组处于停机状态。

表 1 DDPSO 算法优化结果

| 时段 | 负荷/MW | 机组发电量/MW | | | | | | | | | |
|----|-------|----------|-----|-----|-----|-----|----|----|----|----|----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 700 | 455 | 245 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 750 | 455 | 295 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 850 | 455 | 370 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 950 | 455 | 455 | 0 | 0 | 50 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1000 | 455 | 390 | 130 | 0 | 25 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1100 | 455 | 360 | 130 | 130 | 25 | 0 | 0 | 0 | 0 | 0 |
| 7 | 1150 | 455 | 410 | 130 | 130 | 25 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1200 | 455 | 455 | 130 | 130 | 30 | 0 | 0 | 0 | 0 | 0 |
| 9 | 1300 | 455 | 455 | 130 | 130 | 85 | 20 | 25 | 0 | 0 | 0 |
| 10 | 1400 | 455 | 455 | 130 | 130 | 105 | 80 | 25 | 20 | 0 | 0 |
| 11 | 1450 | 455 | 455 | 130 | 130 | 90 | 80 | 25 | 55 | 30 | 0 |
| 12 | 1500 | 455 | 455 | 130 | 130 | 80 | 80 | 50 | 55 | 55 | 10 |
| 13 | 1400 | 455 | 455 | 130 | 130 | 65 | 80 | 30 | 55 | 0 | 0 |
| 14 | 1300 | 455 | 455 | 130 | 130 | 85 | 20 | 25 | 0 | 0 | 0 |
| 15 | 1200 | 455 | 455 | 130 | 130 | 30 | 0 | 0 | 0 | 0 | 0 |
| 16 | 1050 | 455 | 355 | 130 | 85 | 25 | 0 | 0 | 0 | 0 | 0 |
| 17 | 1000 | 455 | 260 | 130 | 130 | 25 | 0 | 0 | 0 | 0 | 0 |
| 18 | 1100 | 455 | 360 | 130 | 130 | 25 | 0 | 0 | 0 | 0 | 0 |
| 19 | 1200 | 455 | 455 | 130 | 130 | 30 | 0 | 0 | 0 | 0 | 0 |
| 20 | 1400 | 455 | 455 | 130 | 130 | 90 | 80 | 25 | 35 | 0 | 0 |
| 21 | 1300 | 455 | 455 | 130 | 130 | 85 | 20 | 25 | 0 | 0 | 0 |
| 22 | 1100 | 455 | 455 | 0 | 0 | 85 | 80 | 25 | 0 | 0 | 0 |
| 23 | 900 | 455 | 395 | 0 | 0 | 0 | 50 | 0 | 0 | 0 | 0 |
| 24 | 800 | 455 | 345 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

图 1 是文献[1]中的 PSO 方法、文献[7]中的 GA 方法和本文的 DDPSO 方法对该算例的收敛比较图。由图可见,DDPSO 算法的收敛曲线下降速度很快,在迭代 150 次左右就已经达到最优值,而 PSO 算法需要迭代 500 次左右,GA 算法需要迭代 300 次左右,说明本文所提算法有较好的收敛性。从图中还可以看出 DDPSO 算法较其他两种方法有更高的计

算精度。

上述三种方法都是随机优化算法,为考查算法的收敛性与求解精度,分别运行三种算法 50 次,三种方法得到的最优运行成本、最差运行成本、平均运行成本和标准差如表 2 所示。由表中的数据可知,DDPSO 算法的优化结果要明显优于其他两种算法,且其标准差仅为 0.000 431,表明该算法具有较好的收敛稳定性。

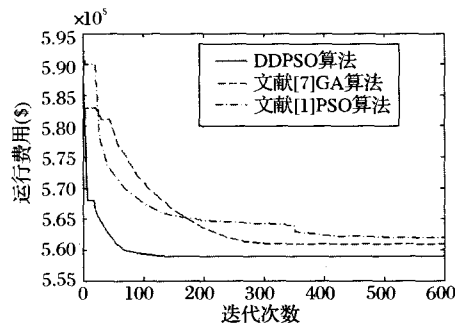


图 1 三种算法的收敛曲线

表 2 不同方法的优化结果

| 方法 | 最优值(\$) | 最差值(\$) | 平均值(\$) | 标准差 |
|-------|---------|---------|---------|----------|
| GA | 560865 | 582416 | 567434 | 0.000675 |
| PSO | 562012 | 585279 | 570814 | 0.000783 |
| DDPSO | 559312 | 568715 | 559997 | 0.000431 |

5 结语

本文将 DDPSO 算法应用于电力系统机组组合问题中,根据机组组合问题的具体特点,采用二维实数矩阵对发电计划安排进行编码,将两层的混合整数规划问题转化为单层的连续优化问题,可以直接使用 DDPSO 算法加以解决,降低了算法的时间复杂度。同时运用个体修正策略处理各项约束条件,使所有粒子均为可行解,也使整个算法始终在可行解区域内寻优,提高了算法的寻优速度和精度。仿真实验的结果也表明了 DDPSO 算法的有效性与鲁棒性。

参考文献:

- [1] GAING Z L. Discrete particle swarm optimization algorithm for unit commitment [C]// Proceedings of IEEE Power Engineering Society General Meeting. Toronto: IEEE Press, 2003: 418-424.
- [2] SUN L Y, ZHANG Y, JIANG C W. A matrix real-coded genetic algorithm to the unit commitment problem [J]. Electric Power Systems Research, 2006, 76(9): 716-728.
- [3] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Perth: IEEE Press, 1995: 1942-1948.
- [4] VASCONCELOS J A, RAMIREZ J A, TAKAHASHI R H C. Improvements in genetic algorithms [J]. IEEE Transactions on Magnetism, 2001, 37(5): 3414-3417.
- [5] OUYANG Z, SHAHIDEHPOUR S M. An intelligent dynamic programming for unit commitment application [J]. IEEE Transactions on Power Systems, 1991, 6(3): 1203-1209.
- [6] ONGSAKUL W, PETCHARAKS N. Unit commitment by enhanced adaptive Lagrangian relaxation [J]. IEEE Transactions Power Systems, 2004, 19(1): 620-628.
- [7] SWARUP K S, YAMASHIRO S. Unit commitment solution methodology using genetic algorithm [J]. IEEE Transactions on Power Systems, 2002, 17(1): 87-91.